

A large, stylized letter 'A' is formed using the characters 'S' and 'Y'. The 'S' characters are arranged in a grid-like pattern to form the left and right sides of the letter, while 'Y' characters form the central vertical stem and the horizontal crossbars. The overall shape is a bold, blocky 'A' that fills most of the page.

[illegible]

(1)	39	HISTORY	; DETAILED
(1)	56	DECLARATIONS	
(1)	78	EXESCLREF	- CLEAR EVENT FLAG SERVICE
(1)	128	SCH\$GETEFC	- GET ADDRESS OF EVENT FLAG CLUSTER
(1)	182	SCH\$CLREF	- CLEAR EVENT FLAG (INTERNAL FORM)
(1)	227	EXESREDEF	- READ EVENT FLAG SYSTEM SERVICE
(1)	271	EXESSETEF	- SET EVENT FLAG SERVICE

```
0000 1      .TITLE SYSEVTSRV - EVENT FLAG SERVICES
0000 2      .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7      *
0000 8      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10     *  ALL RIGHTS RESERVED.
0000 11     *
0000 12     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17     *  TRANSFERRED.
0000 18     *
0000 19     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21     *  CORPORATION.
0000 22     *
0000 23     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25     *
0000 26     *****
0000 27
0000 28     ++
0000 29     FACILITY: EXECUTIVE, EVENT FLAG SERVICES
0000 30
0000 31     ABSTRACT: SYSEVTSRV IMPLEMENTS THE CLEAR READ AND SET EVENT FLAG SYSTEM
0000 32     SERVICES AND CONTAINS THE INTERNAL CLEAR EVENT FLAG ROUTINE.
0000 33
0000 34     --
0000 35
0000 36     VERSION:
0000 37
0000 38     .PAGE
0000 39     .SBTTL HISTORY ; DETAILED
0000 40
0000 41     AUTHOR:
0000 42     R. HUSTVEDT : VERSION
0000 43
0000 44     MODIFIED BY:
0000 45
0000 46     V02-005 SRB0036 Steve Beckhardt 27-Oct-1981
0000 47     Changed SCH$GETEFC to always return via RSB. Changed
0000 48     SCH$CLREF and SCH$READEP to do the RET on error. Added
0000 49     alternate entry point SCH$CLREFR to always return via RSB.
0000 50
0000 51     V02-004 LJK0024 Lawrence J. Kenah 14 May 1981
0000 52     Only take fast exit path out of $SETEP if call to SCH$POSTEP
0000 53     was successful. This insures that system service exceptions
0000 54     work correctly. Introduce symbolic offset for saved FP.
```



```

0000 56 .SBTTL DECLARATIONS
0000 57
0000 58 :
0000 59 : INCLUDE FILES:
0000 60 :
0000 61
0000 62 $CEBDEF ; COMMON EVENT BLOCK DEFINITIONS
0000 63 $DYNDEF ; DYNAMIC DATA STRUCTURE TYPE DEFS
0000 64 $IPLDEF ; IPL DEFINITIONS
0000 65 $PCBDEF ; PCB DEFINITIONS
0000 66 $PRIDEF ; PRIORITY INCREMENT DEFINITIONS
0000 67 $SFDEF ; Stack frame symbolic offsets
0000 68 $SSDEF ; STATUS CODE DEFINITIONS
0000 69 :
0000 70 : EQUATES:
0000 71 :
00000004 0000 72 EFN=4 ; DISPLACEMENT TO EVENT FLAG NUMBER ARG
00000008 0000 73 MASKP=8 ; DISPLACEMENT TO MASK POINTER
0000 74
00000000 0000 75 .PSECT AEXENONPAGED,BYTE ; NONPAGED EXEC
0000 76

```

```
0000 78 .SBTTL EXESCLREF - CLEAR EVENT FLAG SERVICE
0000 79
0000 80 :++
0000 81 : FUNCTIONAL DESCRIPTION:
0000 82 : THE CLEAR EVENT FLAG SYSTEM SERVICE CLEARS THE SPECIFIED
0000 83 : EVENT FLAG. NO RESCHEDULING IS EVER NECESSARY AFTER CLEARING
0000 84 : AN EVENT FLAG.
0000 85 :
0000 86 :
0000 87 : CALLING SEQUENCE:
0000 88 : CALL ARGLIST,EXESCLREF
0000 89 :
0000 90 :
0000 91 : INPUT PARAMETERS:
0000 92 : 04(AP) - EVENT FLAG NUMBER TO BE CLEARED
0000 93 : R4 - PCB ADDRESS OF CURRENT PROCESS
0000 94 :
0000 95 : IMPLICIT INPUTS:
0000 96 :
0000 97 : OUTPUT PARAMETERS:
0000 98 : R0 - COMPLETION STATUS CODE, ERROR IF BIT 0 IS SET.
0000 99 :
0000 100 :
0000 101 : IMPLICIT OUTPUTS:
0000 102 : THE EVENT FLAG SPECIFIED WITHIN THE CLUSTER SPECIFIED
0000 103 : IS CLEARED.
0000 104 :
0000 105 : COMPLETION CODES:
0000 106 : $$$_WASCLR - EVENT FLAG WAS ALREADY CLEAR
0000 107 : $$$_WASSET - EVENT FLAG WAS SET
0000 108 : $$$_ILLEFC - ILLEGAL CLUSTER NUMBER (I.E. 4 - 7)
0000 109 : $$$_UNASEFC - UNASSIGNED CLUSTER NUMBER
0000 110 :
0000 111 : SIDE EFFECTS:
0000 112 : NONE
0000 113 :
0000 114 : --
0000 115 :
0000 116 :
0000 117 :
0000 118 EXESCLREF::
0000 119 .WORD ^M<R2,R3,R4> ; CLEAR EVENT FLAG SERVICE
53 04 AC 001C 0000 120 MOVZBL EFN(AP),R3 ; ENTRY MASK SAVING R2,R3,R4
0006 121 BSBB SCH$CLREF ; GET EVENT FLAG NUMBER
5D 0C AD D0 0008 122 MOVL SF$SAVE_FP(FP),FP ; CALL INTERNAL ROUTINE
SE 00' C0 000C 123 ADDL S^#EXESC_CMSTKSZ,SP ; RESTORE FRAME POINTER
000F 124 REI ; CLEAN STACK BACK TO PC,PSL
0010 125 ; RETURN
0010 126
```



```
0010 128 .SBTTL SCH$GETEFC - GET ADDRESS OF EVENT FLAG CLUSTER
0010 129 :++
0010 130 : FUNCTION:
0010 131 : SCH$GETEFC COMPUTES THE ADDRESS OF THE EVENT FLAG CLUSTER
0010 132 : SELECTED BY THE EVENT FLAG NUMBER IN R3. THE EVENT FLAG CLUSTER
0010 133 : ADDRESS IS RETURNED IN R2.
0010 134 :
0010 135 : CALLING SEQUENCE:
0010 136 : BSBB/BSBW/JSB SCH$GETEFC
0010 137 :
0010 138 : INPUT PARAMETERS:
0010 139 : R3 - EVENT FLAG NUMBER
0010 140 : R4 - PCB ADDRESS
0010 141 :
0010 142 : OUTPUT PARAMETERS:
0010 143 : R0 - STATUS CODE
0010 144 : R1 - IF LOCAL CLUSTER, (0 OR -1)
0010 145 : IF COMMON CLUSTER, (2 OR 1)
0010 146 : R2 - ADDRESS OF EVENT FLAG CLUSTER
0010 147 : R3 - EVENT FLAG NUMBER WITHIN CLUSTER
0010 148 :
0010 149 : COMPLETION CODES:
0010 150 : $$$_NORMAL - NORMAL COMPLETION STATUS
0010 151 : $$$_ILLEFC - ILLEGAL CLUSTER NUMBER
0010 152 : $$$_UNASEFC - UNASSIGNED CLUSTER NUMBER
0010 153 :
0010 154 : --
0010 155 :
0010 156 SCH$GETEFC::
0010 157 : GET ADDRESS OF EVENT FLAG CLUSTER
51 53 50 01 3C 0010 157 MOVZWL #$$$_NORMAL,R0 : ASSUME NORMAL COMPLETION
53 53 03 05 EE 0013 158 EXTV #5,#3,R3,R1 : EXTRACT CLUSTER NUMBER
52 50 A441 DE 001A 159 BLSS 30$ : ILLEGAL WITH 4,5,6,7
53 FFFFFFFE0 8F CA 001F 160 MOVAL PCB$EFC$(R4)[R1],R2 : GET ADDR OF EF VECTOR
01 51 F5 0026 161 BICL #^C<^X1F>,R3 : EXTRACT EVENT NUMBER WITHIN CLUSTER
05 0029 162 SOBGTR R1,10$ : BR IF COMMON
52 62 10 C1 002A 163 RSB : RETURN WITH SUCCESS
15 18 002E 164 10$: ADDL3 #CEB$L_EFC,(R2),R2 : GET POINTER TO CEB EVENT MASK
FA A2 2D 91 0030 165 BGEQ 40$ : BR IF NOT ASSIGNED
01 13 0034 166 CMPB #DYN$C_SLAVCEB,<CEB$B_TYPE-CEB$L_EFC>(R2) : IS THIS A SLAVE CEB?
05 0036 167 BEQL 20$ : BR IF SLAVE, FIND MASTER CEB ADR
0037 168 RSB : RETURN WITH SUCCESS
0037 169 :
0037 170 : COMPUTE ADDRESS OF MASTER CEB IN SHARED MEMORY.
52 30 A2 D0 0037 171 20$: MOVL <CEB$L_MASTER-CEB$L_EFC>(R2),R2 : GET ADR OF SH MEM MASTER CEB
52 10 C0 003B 172 ADDL2 #CEB$L_EFC,R2 : POINT TO EFC IN MASTER CEB
05 003E 173 RSB
003F 174
50 00EC 8F 3C 003F 175 30$: MOVZWL #$$$_ILLEFC,R0 : SET ERROR CODE FOR ILLEGAL EFC
05 0044 176 RSB : AND RETURN
0045 177
50 0234 8F 3C 0045 178 40$: MOVZWL #$$$_UNASEFC,R0 : SET ERROR CODE FOR UNASSIGNED
05 004A 179 RSB : AND RETURN
004A 180
```

```
004B 182 .SBTTL SCH$CLREF - CLEAR EVENT FLAG (INTERNAL FORM)
004B 183 :++
004B 184 SCH$CLREF - CLEAR EVENT FLAG, INTERNAL FORM
004B 185 :
004B 186 CALLING SEQUENCE:
004B 187 JSB/BSB SCH$CLREF
004B 188 :
004B 189 SCH$CLREF WILL RETURN VIA RET RATHER THAN THE NORMAL RSB ON ERROR
004B 190 CONDITIONS
004B 191 SCH$CLREFR IS AN ALTERNATE ENTRY POINT THAT ALWAYS RETURNS VIA RSB.
004B 192 :
004B 193 INPUT:
004B 194 R3 - EVENT FLAG NUMBER, ZERO FILLED.
004B 195 R4 - PCB ADDRESS
004B 196 :
004B 197 OUTPUT:
004B 198 R0 - COMPLETION STATUS, ERROR IF BIT 0 CLEAR.
004B 199 :
004B 200 COMPLETION CODES:
004B 201 SSS_WASCLR - EVENT FLAG WAS ALREADY CLEAR
004B 202 SSS_WASSET - EVENT FLAG WAS SET
004B 203 SSS_ILLEFC - ILLEGAL CLUSTER NUMBER
004B 204 SSS_UNASEFC - UNASSIGNED CLUSTER NUMBER
004B 205 :--
004B 206 :
004B 207 .ENABL LSB
004B 208 :
004B 209 SCH$CLREFR:: : CLEAR EVENT FLAG (AND RETURN TO CALLER)
004B 210 BSBB SCH$GETEFC : GET ADDRESS OF EVENT FLAG CLUSTER
06 50 10 004B 211 BLBS R0,5$ : ON SUCCESS JOIN COMMON CODE
05 0050 212 RSB : RETURN ON FAILURE
0051 213 :
0051 214 SCH$CLREF:: : CLEAR EVENT FLAG INTERNAL
0051 215 BSBB SCH$GETEFC : GET ADDRESS OF EVENT FLAG CLUSTER
08 50 10 0051 216 BLBC R0,20$ : BRANCH IF ERROR
0053 217 ASSUME SSS_WASCLR EQ SSS_NORMAL
0056 218 :
0056 219 5$: BBCCI R3,(R2),10$ : SSS_NORMAL RETURNED BY SCH$GETEFC
03 62 53 0056 220 MOVL #SSS_WASSET,R0 : CLEAR AND TEST FLAG
50 09 05 005A 221 10$: RSB : REPORT STATUS
005D 222 : RETURN TO CALLER
005E 223 20$: RET : EXIT SERVICE IF ERROR
005F 224 :
005F 225 .DSABL LSB
```



```
005F 227 .SBTTL EXES$READEF - READ EVENT FLAG SYSTEM SERVICE
005F 228 :++
005F 229 : FUNCTIONAL DESCRIPTION:
005F 230 : EXES$READEF IMPLEMENTS THE READ EVENT FLAG SYSTEM SERVICE WHICH
005F 231 : RETURNS THE STATE OF THE EVENT FLAG CLUSTER SELECTED BY THE
005F 232 : SPECIFIED EVENT FLAG NUMBER.
005F 233 :
005F 234 : CALLING SEQUENCE:
005F 235 : CALLG  ARGLIST,EXES$READEF
005F 236 :
005F 237 : INPUT PARAMETERS:
005F 238 : 04(AP) - EVENT FLAG NUMBER TO SELECT PROPER CLUSTER
005F 239 : 08(AP) - ADDRESS TO STORE STATE OF EVENT FLAG CLUSTER
005F 240 : R4 - PCB ADDRESS OF CURRENT PROCESS
005F 241 :
005F 242 : OUTPUT PARAMETERS:
005F 243 : R0 - COMPLETION STATUS CODE
005F 244 : @08(AP) - STATE OF 32 EVENTS IN SPECIFIED CLUSTER
005F 245 :
005F 246 : COMPLETION CODES:
005F 247 : SSS_WASSET - SPECIFIC EVENT FLAG IS CURRENTLY SET
005F 248 : SSS_WASCLR - SPECIFIC EVENT FLAG IS CURRENTLY CLEAR
005F 249 : SSS_ILLEFC - ILLEGAL EVENT FLAG CLUSTER NUMBER
005F 250 : SSS_UNASEFC - UNASSIGNED EVENT FLAG CLUSTER
005F 251 : SSS_ACCVIO - ACCESS VIOLATION ON WRITE DESTINATION
005F 252 :
005F 253 :--
005F 254
005F 255 EXES$READEF::
005F 256 .WORD ^M<R2,R3,R4> : READ EVENT FLAG SYSTEM SERVICE
53 04 AC 9A 0061 257 MOVZBL EFN(AP),R3 : ENTRY MASK SAVING R2,R3,R4
18 50 E9 0065 258 BSBB SCH$GETEFC : GET EVENT FLAG NUMBER
0067 259 BLBC R0,20$ : GET EFC ADDRESS AND CONTENT
006A 260 ASSUME SSS_WASCLR EQ SSS_NORMAL : BRANCH ON ERROR
51 08 AC D0 006A 261 MOVL MASKP(AP),R1 : GET ADDRESS OF DESTINATION
006E 262 IFNOWRT #4,(R1),10$ : TEST FOR WRITABLE
07 61 62 D0 0074 263 MOVL (R2),(R1) : STORE EVENT FLAG BIT VECTOR
07 61 53 E1 0077 264 BBC R3,(R1),20$ : EXIT IF CLEAR
50 09 3C 007B 265 MOVZWL #SS$_WASSET,R0 : RETURN SET STATUS CODE
007E 266 RET : AND RETURN
50 0C 3C 007F 267 10$: MOVZWL #SS$_ACCVIO,R0 : SET ACCESS VIOLATION
0082 268 20$: RET : AND RETURN
0083 269
```

```
0083 271 .SBTTL EXESSETEF - SET EVENT FLAG SERVICE
0083 272 :++
0083 273 : FUNCTIONAL DESCRIPTION:
0083 274 : EXESSETEF IMPLEMENTS THE SET EVENT FLAG SERVICE WHICH SETS
0083 275 : THE SPECIFIED EVENT FLAG AND CAUSES ANY APPROPRIATE RESCHEDULING.
0083 276 :
0083 277 : CALLING SEQUENCE:
0083 278 : CALLG EXESSETEF
0083 279 :
0083 280 : INPUT PARAMETERS:
0083 281 : 04(AP) - EVENT FLAG NUMBER
0083 282 : R4 - PCB ADDRESS OF CURRENT PROCESS
0083 283 :
0083 284 : IMPLICIT INPUTS:
0083 285 : CURRENT PCB LOCATED VIA SCH$GL_CURPCB
0083 286 :
0083 287 : OUTPUT PARAMETERS:
0083 288 : R0 - COMPLETION STATUS CODE
0083 289 :
0083 290 : COMPLETION CODES:
0083 291 : SSS_WASCLR - SPECIFIED EVENT WAS CLEAR INITIALLY
0083 292 : SSS_WASSET - SPECIFIED EVENT WAS SET INITIALLY
0083 293 : SSS_ILLEFC - ILLEGAL CLUSTER NUMBER
0083 294 : SSS_UNASEFC - UNASSIGNED EVENT FLAG CLUSTER NUMBER
0083 295 : SSS_BADQUEUEHDR - UNABLE TO MAKE PROCESSOR REQUEST TO UPDATE SLAVE CEB
0083 296 :
0083 297 :--
0083 298 :
0083 299 EXESSETEF::
0083 300 .WORD ^M<R2,R3,R4,R5> ; SET EVENT FLAG SYSTEM SERVICE
0083 301 MOVZBL EFN(AP),R3 ; ENTRY MASK SAVING R2,R3,R4,R5
0083 302 MOVL PCB$P_PID(R4),R1 ; GET EVENT FLAG NUMBER
0083 303 MOVL #PRI$IOCOM,R2 ; GET PROCESS ID (PID)
0083 304 BSBW SCH$POSTEF ; SET PRIORITY INCREMENT CLASS
0083 305 BLBC R0,10$ ; POST EVENT FLAG
0083 306 MOVL SF$SAVE_FP(FP),FP ; DON'T USE FAST EXIT IF ERROR
0083 307 ADDL S^#EXESC_CMSTKSZ,SP ; RESTORE FRAME POINTER
0083 308 REI ; CLEAN STACK BACK TO PC,PSL
0083 309 ; AND RETURN
0083 310 10$: RET ; EXIT THROUGH CHANGE MODE DISPATCHER
0083 311 ; COMMON EXIT PATH IF ERROR
0083 312
0083 313 .END
```

53 04 AC 9A 003C 0083 300
51 60 A4 D0 0085 301
52 01 D0 0089 302
FF6D' 30 008D 303
08 50 E9 0090 304
SD 0C AD D0 0093 305
5E 00' C0 0096 306
02 009A 307
009D 308
009E 309
04 009E 310 10\$: RET
009F 311
009F 312
009F 313

SYSEVTSRV
Symbol table

- EVENT FLAG SERVICES

I 6

16-SEP-1984 02:04:29 VAX/VMS Macro V04-00
5-SEP-1984 03:53:06 [SYS.SRC]SYSEVTSRV.MAR;1Page 8
(1)

```
CEBSB_TYPE      = 0000000A
CEBSL_EFC       = 00000010
CEBSL_MASTER    = 00000040
DYN$C_SLAVCEB   = 0000002D
EFN             = 00000004
EXESCLREF       = 00000000 RG 02
EXESC_CMSTKSZ   = ***** X 02
EXESREADEF      = 0000005F RG 02
EXESSETEF       = 00000083 RG 02
MASKP          = 00000008
PCBSL_EFCS      = 00000050
PCBSL_PID       = 00000060
PRIS_TOCOM      = 00000001
SCH$CLREF       = 00000051 RG 02
SCH$CLREFR      = 0000004B RG 02
SCH$GETEFC      = 00000010 RG 02
SCH$POSTEF      = ***** X 02
SFSL_SAVE_FP    = 0000000C
SS$_ACCVIO      = 0000000C
SS$_ILLEFC      = 000000EC
SS$_NORMAL      = 00000001
SS$_UNASEFC     = 00000234
SS$_WASCLR      = 00000001
SS$_WASSET      = 00000009
```

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR
AEXENONPAGED	0000009F (159.)	02 (2.)	NOPIC USR

CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:01.21
Command processing	106	00:00:00.58	00:00:05.47
Pass 1	266	00:00:07.23	00:00:28.11
Symbol table sort	0	00:00:01.10	00:00:03.18
Pass 2	68	00:00:01.42	00:00:03.41
Symbol table output	4	00:00:00.05	00:00:00.49
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	477	00:00:10.48	00:00:41.89

The working set limit was 1350 pages.
40900 bytes (80 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 780 non-local and 10 local symbols.
313 source lines were read in Pass 1, producing 13 object records in Pass 2.
16 pages of virtual memory were used to define 15 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

\$255\$DUA28:[SYS.OBJ]LIB.MLB;1
\$255\$DUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)

6
6
12

866 GETS were required to define 12 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SYSEVTSRV/OBJ=OBJ\$:SYSEVTSRV MSRC\$:SYSEVTSRV/UPDATE=(ENH\$:SYSEVTSRV)+EXECML\$/LIB

0384 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

